TASK FORCE WEB

Web Service Execution Engine (WSEE)

9 July 2002

D. Taylor Holmes, P.D.I.T.
Senior Engineer
SPAWAR Systems Center Charleston

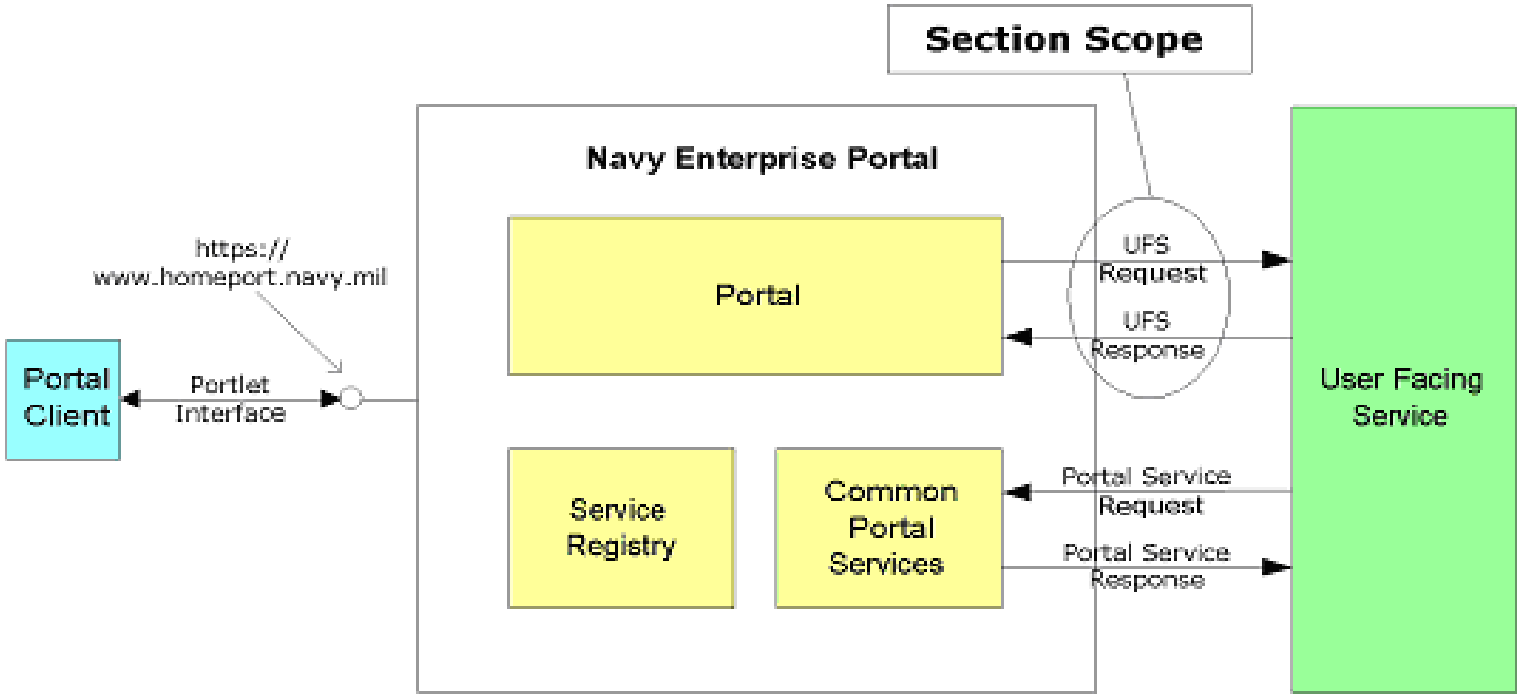
Agenda

- The Web Services Vision
- Web Service Execution Engine (WSEE) Overview
  - UFS Interface, UFS Request Revisited
  - WSEE 1.0 Request Interface
  - Example WSEE Request
  - A Dynamic SOAP Client
  - UFS Response Revisited
- End-to-End WSEE Demonstration

The Web Services Vision

- The Web Services model offers an exciting opportunity to connect disparate software applications and produce valuable new information systems.
- When application business logic and data is exposed as Web Services, developers will be empowered to overcome existing information barriers.
- The NEP Web Services effort can help produce a better informed, more powerful and connected Navy.

WSEE Overview - UFS Interface Revisited



WSEE Overview – UFS Request Revisited

- The NEP will generate 2 primary types of User Facing Service Requests based on the bindingType of the registered service:
  - HTTP Request -> HTTP-Bound UFS
  - HTTP|SOAP Request -> WSDL-Bound UFS
- The Web Service Execution Engine (WSEE) will be responsible for dynamically generating the HTTP|SOAP Request to the WSDL-Bound UFS

WSEE Overview – WSEE 1.0 Request Interface

- WSEE 1.0 Request Interface
  - bindingKey (required) – a valid bindingKey GUID from a UDDI Service Registry
  - operation (required) – the operation defined in the Web Service's WSDL Document that you wish to execute.
  - operation message parts – the input parameters required to invoke the Web Service operation's input message. Passed to the WSEE via standard HTTP form POST or Query String parameters.
  - uddiURL (optional) – the URL to the UDDI Registry that holds the Web Service registration. Defaults to the NEP Service Registry.

WSEE Overview – example WSEE Request

```
<HTML>
<HEAD>
    <TITLE>Web Service Execution Engine Tester</TITLE>
</HEAD>
<BODY><P><BR></P>
    <H3>A simple HTML Form POST to the WSEE that will dynamically bind to the
  Service Registry UDDI API and call the get_businessDetail operation</H3>
    <FORM name="frmget_businessDetail" action="http://homeport.navy.mil/servlet/WSExecute"
        method="post">
        <DIV style="DISPLAY: none">
            <INPUT value="78B2CCC9-D2C4-44A7-9602-18915E0B9811" name="bindingKey">
            <INPUT value="get_businessDetail" name="operation">
        </DIV>
        <TABLE>
            <TR><TH>get_businessDetail Web Service Call</TH></TR>
            <TR><TD>business Key</TD><TD>
            <INPUT size="60" value="8AE56BB5-0916-49E0-A359-9E69A7D0C8CB"
                name="businessKey">
            </TD></TR>
            <TR><TD><INPUT id="submit1" type="submit" value="Invoke"
                name="submit1"></TD></TR>
        </TABLE>
    </FORM>
<P><BR></P></BODY>
</HTML>
```

WSEE Overview – A Dynamic SOAP Client

- The Web Service Execution Engine (WSEE) solves the functional problem of hard binding to SOAP-based Web Services.
  - Frees application developers from SOAP Client details and having to code to a specific SOAP interface
  - Dynamically builds SOAP Client message based on registered WSDL Document

WSEE Overview – A Dynamic SOAP Client

- The WSEE will dial up the WSDL referenced in the Service Registry and will dynamically generate the SOAP Client Message
  - Show WSDL example
- The WSEE will send the SOAP Message to the Service Access Point and receive a SOAP Response or SOAP Fault Message

WSEE Overview – UFS Response Revisited

- The User Facing Web Service is responsible for returning a valid SOAP Response message.
  - Show example of SOAP Response Message
- The User Facing Web Service is responsible for returning a SOAP Fault Message if an error occurs.
  - Show example of SOAP Fault Message

End-to-End WSEE Demonstration

- Show an example SOAP-based User Facing Web Service
- Show the WSDL Document for this example
- Show the Portlet registration for this example
- Show initial and subsequent invocation through WSEE